Web Page Summarization via HTML Tables

Minoru YOSHIDA, Jun'ichi TSUJII
Department of Information Science
Graduate School of Science
University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-0033, Japan
{mino,tsujii}@is.s.u-tokyo.ac.jp

Paper ID:

Keywords: Summarization, WWW, Ontologies, HTML tables

Contact Author: Minoru YOSHIDA
Department of Information Science
Graduate School of Science
University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-0033, Japan
mino@is.s.u-tokyo.ac.jp

Under consideration for other conferences (specify)? ACL02 Student Research Workshop (submitted)

Abstract

In this paper we propose a new algorithm for summarizing Web pages. Our algorithm extracts essential parts of Web pages based on a Hidden Markov Model (HMM). The key idea in our algorithm is to estimate parameters for the HMM by consulting HTML tables on the WWW. We present some techniques required for estimating the parameters. In the series of experiments we show that our HMM model outperformed a simple Naive Bayes one.

Web Page Summarization via HTML Tables

Paper ID:

Abstract

In this paper we propose a new algorithm for summarizing Web pages. Our algorithm extracts essential parts of Web pages based on a Hidden Markov Model (HMM). The key idea in our algorithm is to estimate parameters for the HMM by consulting HTML tables on the WWW. We present some techniques required for estimating the parameters. In the series of experiments we show that our HMM model outperformed a simple Naive Bayes one.

1 Introduction

The rapid growth of the WWW in recent years has drastically increased the amount of information available on the Web. However, it also made it difficult for users to find, among a lot of pages, the appropriate ones they are searching. This is also the case even if a search engine is employed because the number of pages in searching results still remain large in many cases and it is often the case highly-ranked pages do not satisfy the user's requirement. Summarizing Web pages is one of the promising ways to solve this problem. If the content of pages is reorganized into some concise representation, and presented with pages or searching results, it will be a great help to users to find the pages of his/her interest.

The algorithm proposed in this paper extracts the parts of Web pages which are essential for the content of the pages. Figure 1 shows an example of such extraction. In more detail, it divides a Web page into several fragments called *blocks*, and then, selects essential blocks and assembles them into a summary. Whether a block is essential or not is determined in a statistical way based on the frequencies of words appearing in a training corpus.

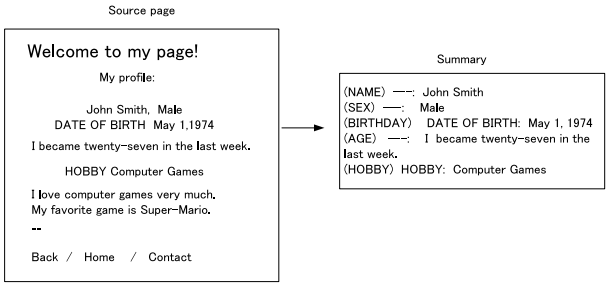

Figure 1: Example of Web page summarization.

We need therefore some criteria to determine which parts of Web pages are essential and which parts are redundant. Our algorithm consults a set of *HTML tables* to make this decision. An HTML table often can be seen as a template and its slots for some scenario. (In this paper we call each slot *an attribute*, and each slot filler *a value*.) For example, about-me pages often contain tables consisting of templates with attributes such as *name*, *sex* and *age*. Our idea is to use such templates and their typical slot-fillers as a training corpus to estimate the essentiality of blocks on the assumption that attributes given in many tables are essential for Web page's content. Moreover, using tables as a training corpus makes it possible to make use of relations between attributes and values, such as "the word *male* tends to appear near the word *sex*," to improve resulting summaries.

There have been some researches on extracting ontologies from HTML tables on the Web (Chen et al., 2000; Yoshida et al., 2001). In this study we assume that an ontology is a description of some class of objects described by attribute-value pairs as exemplified in Figure 2. Ontologies extracted by their methods are used as a training corpus to estimate words' relevancy to the essential content of Web pages.

```
[ATTRIBUTE] ------ [POSSIBLE VALUE]

    {Name} ---------{John Smith,
                     Ichiro Suzuki,
                     Mary, ...}

{Gender, Sex}---------{Male, Female, ...}

 {Bloodtype, ---------{A, AB, Rh-O,...}
       Blood}
```

Figure 2: An example ontology of human. Each attribute or value is represented by a bag of strings expressing it.

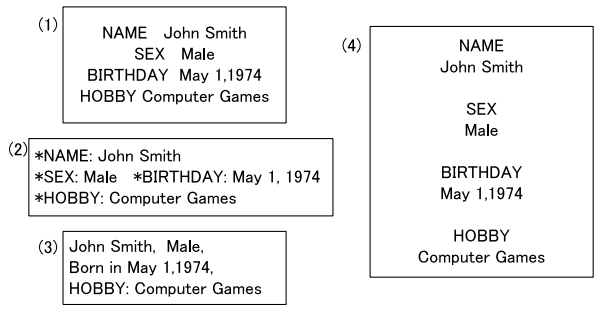

Figure 3: Various formats for the same contents. There is the variety of attribute indicator. In some cases attributes are entirely omitted.

Notice that their algorithms are domain independent and need no training samples labeled by hand. We can thus apply our algorithm to various topics with no human efforts to label training samples.

In addition, our algorithm not only selects relevant blocks as a summary, but also assigns the appropriate category to each block, because states in the HMM correspond to categories such as *NAME* or *SEX*. It makes it easy to use resulting summaries for further applications, such as making a digest of several Web pages.

The remainder of this paper is organized as follows. In Section 2, we give the term definitions. In Section 3, we will explain our system in detail. Section 4 shows some preliminary experiments and Section 5 concludes this paper and shows future directions of this work.

2 Assumptions and Definitions

In this section we first give some assumptions on Web pages. After that, we give definition of the terms used in the remainder of this paper.

2.1 Web Pages as Block Sequences

Web pages consist of not only sentences, but also *non-sentential blocks* such as tables and lists. Figure 3 shows example pages consisting of non-sentential blocks. Although there are various types of expressions even for the same content, they can be treated uniformly in the form of *a sequence of blocks* separated by HTML tags or other special characters. Block sequences can be seen as the variant of text documents (i.e., a sequence of sentences) because blocks and sentences are the same in that both of them can be seen as bags of words. Therefore traditional statistical algorithms which have been successfully applied to text summarization methods (Kraaij et al., 2001; Conroy et al., 2001) also could be applied to Web page summarization. Among existing statistical models, we chose Hidden Markov Models (HMMs) because neighboring blocks are often closely related like *SEX* and *Male* in the example pages in Figure 3.

2.2 Term Definition

In the following we give definitions of the terms used in this paper.

- A *page* is a sequence of *page fragments*, each of which is either *a block* or *a separator*.
- A block b is a bag of words.
- A separator is a sequence of HTML tags or *special characters*. The special characters are characters which tend to be used as boundaries of blocks. They are defined a priori.
- An *ontology* is a sequence $\langle (A_1, V_1), (A_2, V_2), ..., (A_m, V_m) \rangle$, where A_i and V_i correspond to the ith attribute in the ontology and its value, respectively. A_i is a *bag* of the strings used in expressing the ith attribute and V_i is that used in expressing its value.

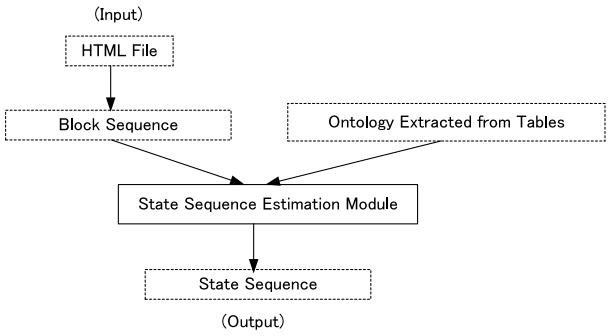

Figure 4: System overview. The SSEM, the main module of the system, uses the ontology extracted from tables.

- A *role* is a pair (l, i), where $l \in \{att, val\}$ and $i \in \{1, 2, ..., m\}$. l, or *a label*, denotes whether a block represents an attribute or a value, and i, or *an index*, denotes the attribute's (or value's) number in the ontology.
- A *state* is defined for each block and has a role as its value. We denote the label of the state s by $l(s)$ and the index by $i(s)$.

3 System Overview

In this section we introduce our system and its algorithm in detail. Figure 4 shows the overall workflow of our system. A Web page given as an input to our system is first decomposed into a sequence of blocks bounded by separators. The State Sequence Estimation Module (SSEM) determines a sequence of states for the block sequence, by using an ontology extracted from HTML tables. Notice that roles for blocks which do not contain any word in ontologies can be estimated from neighboring states thanks to an HMM.

There is the following problem to be solved through this process.

Combining separated but semantically-continuing parts Sometimes one role crosses over two or more neighboring blocks because separators are sometimes used only for the purpose of adjusting looks of a page. For example, the page (1) in Figure 3 uses a space as a separator, which separates values such as *John Smith* into more than one blocks. We introduce a model of *boundaries of roles* to solve this problem as described in Section 3.4.

In the remainder of this section, we briefly explain ontology extraction from tables first. After that, a method to estimate a sequence of states is described.

3.1 Ontology Extracted from Tables

Our system uses an ontology extracted from HTML tables. There have been some researches on the task of extracting ontologies from HTML tables on the WWW (Chen et al., 2000; Yoshida et al., 2001). In this research we used the algorithm described by Yoshida et al. (2001) to extract ontologies. Notice that we used only top 10 attributes (and their values) as roles to be contained in resulting summaries where attributes are ranked in order of their frequency (i.e., $|A_i|$). Each ontology output by this algorithm has a form defined in the previous section: a sequence of pairs of bag of strings. We made a bag of words from each bag of strings in an ontology by decomposing each string into words by a Japanese morphological analyzer JUMAN (Kurohashi and Nagao, 1998). From each bag of words, a frequency of a word-role pair $C(w, r)$ is enumerated as the number of times that w appears in r. This $C(w, r)$ is the base of calculation of probabilities in the SSEM.

3.2 State Sequence Estimation Module

Given a sequence of blocks $\mathcal{B} = \langle b_1, b_2, ..., b_n \rangle$, the State Sequence Estimation Module (SSEM) estimates the most probable sequence of states $\mathcal{S} = \langle s_1, s_2, ..., s_n \rangle$. Here s_i is a state given to the block b_i.

The SSEM estimates the $\mathcal{S}$ so that $P(\mathcal{S}|\mathcal{B})$ takes the highest value. In other words, $\mathcal{S}$ is estimated according to the following formula.

$$\begin{aligned} \hat{\mathcal{S}} &= \arg\max_{\mathcal{S}} P(\mathcal{S}|\mathcal{B}) \\ &= \arg\max_{\mathcal{S}} \frac{P(\mathcal{S}, \mathcal{B})}{P(\mathcal{B})} \\ &= \arg\max_{\mathcal{S}} P(\mathcal{S}, \mathcal{B}) \end{aligned}$$

We said that we used an HMM as a model for $\mathcal{B}$. $P(\mathcal{S}, \mathcal{B})$ is thus approximated in the following form.

$$P(\mathcal{S}, \mathcal{B}) \approx \prod_{i=1}^{n} P(s_i|s_{i-1})P(b_i|s_i) \qquad (1)$$

where $P(s_1|s_0)$ is defined as $P(s_1|s_0) = P(s_1)$. This approximation assumes that a probability for a state is effected only by the previous state, and a probability for the ith block is effected only by the ith state. Based on this formula, the most probable sequence $\mathcal{S}$ is calculated by the standard Viterbi algorithm (Jelinek, 1998).

In the following sections, we explain how to estimate values of $P(b|s)$ and $P(s_i|s_{i-1})$, both of which are needed to calculate the above probability.

3.3 Estimation of $P(b|s)$

Because a block is a bag of words, the probability $P(b|s)$ is calculated as $\prod_{w \text{ in } b} P(w|s)$. Each $P(w|s)$ is defined as

$$\frac{C(w,s)}{C(s)} = \frac{C(w,s)}{\sum_{x \in W} C(x,s)}$$

where W is a set of all the words appearing in the ontology. The main problem is that there are many pairs with zero frequencies, that is, the pairs (w, s) such that $C(w, s) = 0$. It is rare that, for some state s, all the words in a block have non-zero value of $C(w, s)$. For this reason, it will be problematic to use just this definition of $P(w|s)$ because most probabilities become zero.

Currently we use the Good-Turing Estimation (Manning and Schütze, 1999) to solve this problem. In Good-Turing Estimation, the frequency r is adjusted as

$$t^* = (t+1)\frac{N_{t+1}}{N_t}$$

where N_t is the number of the kinds of pairs (w, s) which occurred just t times in the training data. In practice, this adjusted frequency is used only for the t such that $t \leq k$. (k is a threshold and currently set to 5.)

Another important point is that our algorithm makes the following special roles to filter out the blocks that do not represent any attribute or value.

Rare role As mentioned in the beginning of this section, we used only top 10 attributes and their values. Other attributes (and their values) are classified as *rare roles* [1]. Among them, the roles in the form (att, i) are classified into *rare attribute role* and those in the form (val, i) are classified into *rare value role*. Blocks with these roles are excluded from resulting summaries. Because these rare roles are made from various kinds of roles, their word distributions are averaged and have no special characteristics. This gathering of low-frequency roles makes it possible to filter out blocks which have no peculiar distribution of words because such blocks are likely to be given with the rare roles rather than any other particular role.

Sentence role If a string in an ontology has at least one period, question mark or exclamation mark, words in the string are classified into the *sentence* role. A block with the words likely to appear in the sentences such as conjunctions or auxiliary verbs tend to be classified into this state. It contributes to filtering out sentences which do not represent any role.

3.4 Estimation of $P(s_i|s_{i-1})$

We use a unigram probability $P(s_i)$, defined as

$$\frac{\sum_w C(w, s_i)}{\sum_s \sum_w C(w, s)},$$

for the basis of calculation of $P(s_i|s_{i-1})$, but the probability is adjusted according to the following heuristics.

- An attribute must be followed by its value.
- A state is likely to be followed by the same state.

The former heuristic is expressed by the following constraint.

[1] We call other (top 10) roles *non-rare roles* Notice that it does not include the sentence role described below.

Constraint-1 If $s_i \neq s_{i-1}$ and $l(s_{i-1}) = att$, s_i must be $(val, i(s_{i-1}))$.

The latter heuristic reflects a problem of *boundaries of roles*. Although a block is a unit of roles in Web pages, it is not ensured that one role corresponds to just one block. Rather, two or more neighboring blocks often play the same role. (See Figure 5.) The algorithm needs boundaries of roles besides those of page blocks to recognize attributes and their values accurately.

To model boundaries of roles, let us encode them by a sequence of bits. In this sequence, a bit in the ith position means whether a block is the end of a role (1), or not (0). (See Figure 5.) We assume that all patterns of boundaries (which are equivalent to all patterns of the bit sequences) have the same probability. The probability of every bit sequence d is therefore given as $\frac{1}{2^{|d|}}$ where $|d|$ is the length of d.

In this model, the probability $P(\mathcal{S}, \mathcal{B})$ is revised as follows.

$$\begin{aligned}
P(\mathcal{S},\mathcal{B}) &\approx \sum_d P(d) \prod_{i=1}^{n} P(s_i|s_{i-1},d) P(b_i|s_i) \\
&\approx \sum_d \frac{1}{2^{|d|}} \prod_{i=1}^{n} P(s_i|s_{i-1},d_{i-1}) P(b_i|s_i) \\
&= \sum_d \prod_{i=1}^{n} \frac{1}{2} P(s_i|s_{i-1},d_{i-1}) P(b_i|s_i) \\
&= \prod_{i=1}^{n} \sum_{d_{i-1}} \{\frac{1}{2} P(s_i|s_{i-1},d_{i-1})\} P(b_i|s_i) \\
&= \prod_{i=1}^{n} \{\frac{1}{2} P(s_i|s_{i-1},d_{i-1}=0) \\
&\quad + \frac{1}{2} P(s_i|s_{i-1},d_{i-1}=1)\} P(b_i|s_i)
\end{aligned}$$

where $P(s_1|s_0, d_0) = P(s_1)$. The last expression has the same form as that derived from the equation 1 by replacing the $P(s_i|s_{i-1})$ with the expression

$$\frac{1}{2} P(s_i|s_{i-1}, d_{i-1}=0) + \frac{1}{2} P(s_i|s_{i-1}, d_{i-1}=1).$$

So we use this expression as the revised definition of $P(s_i|s_{i-1})$.

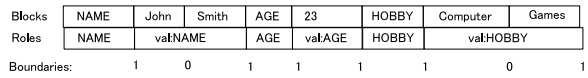

Figure 5: Blocks, Roles, and Boundaries as a bit sequence

We set the following constraint to $P(s_i|s_{i-1}, d_{i-1})$ corresponding to the second heuristic.

Constraint-2 $P(s_i|s_{i-1}, d_{i-1})$ must be 0 if $d_{i-1} = 0$ and $s_i \neq s_{i-1}$, or if $d_{i-1} = 1$ and $s_i = s_{i-1}$.

It leads to the formula $P(s_i|s_{i-1}) = \frac{1}{2}$ for $s_i = s_{i-1}$. (It corresponds to the case when $d_{i-1} = 0$.) The remaining probabilities (for the case when $d_{i-1} = 1$) are distributed among other values of s_i according to the unigram probabilities $P(s_i)$. As the consequence we derive the formula

$$P(s_i|s_{i-1}) = \frac{1}{2} \cdot \frac{P(s_i)}{\sum_{s \neq s_{i-1}} P(s)}$$

when $s_i \neq s_{i-1}$ and $l(s_{i-1}) = val$, and

$$\begin{aligned}
P(s_i|s_{i-1}) &= \frac{1}{2} \quad \text{if } i(s_i) = i(s_{i-1}), \\
&\qquad\qquad l(s_i) = val \\
P(s_i|s_{i-1}) &= 0 \quad \text{otherwise}
\end{aligned}$$

when $l(s_{i-1}) = att$ (by the Constraint-1).

4 Preliminary Experiments

We gathered 10 about-me pages[2] from the WWW to evaluate our system, and decomposed them into blocks. These pages did not include any `<table>` tag. All the blocks that have non-rare roles (75 blocks in total) were extracted and labeled with the correct roles. We picked up one ontology describing a human entity among the ones extracted from tables[3]. Performance was evaluated by comparing the roles output by our algorithm with the ones labeled by hand. Precision is given by N_c/N_m and recall is given by

[2]All these pages were written in Japanese.

[3]All these tables were written in Japanese.

Method	Rec.	Prec.	F-measure
HMM	0.73	0.55	0.63
Naive Bayes	0.65	0.56	0.60
HMM (w/o sent.)	0.80	0.34	0.48
Naive Bayes (w/o sent.)	0.69	0.40	0.51

Table 1: Results in recall and precision. Rec. means recall and Prec. means precision.

N_c/N_h where N_m is the number of roles output by the algorithm, N_h is the number of roles given by hand and N_c is the number of roles output correctly F-measure was calculated as follows.

$$F = \frac{2 \cdot recall \cdot precision}{recall + precision}$$

We also evaluated the accuracy when Naive Bayes Classifier (Manning and Schütze, 1999) was used. The result for Naive Bayes Classifier can be seen as the result when dependency between states was not used in the HMM.

Table 1 shows the result. "W/o sent." means the results when no sentence state was used. The HMM using the sentence state achieved the highest performance in F-measure. It would be a strong evidence for usefulness of the sentence state and dependency between states. Introducing of the sentence state improved precision because it contributed to filter out useless sentences (i.e., sentences without non-rare roles.) However, some necessary blocks (blocks with non-rare roles) were also filtered out and thus precision decreased to some extent. Some revision of definition for the sentence state would be needed to improve precision. The HMM outperformed the Naive Bayes when the sentence state was used although it was not the case when that state was not used. This was because there were many successive sentences, which occupied some neighboring blocks.

5 Conclusion and Future Work

In this paper we proposed a method to summarize Web pages by consulting ontologies extracted from HTML tables. We employed Good Turing Estimation in calculating probabilities, and introduced a sentence role to filter out useless blocks in a Web page. We applied an HMM as a model for calculating of probabilities for block sequences. We showed that the HMM performed better than the Naive Bayes Classifier.

Because the task proposed in this paper is still simple one, it can be extended in several directions. For example, some topic detection techniques might be employed to deal with Web documents on many kinds of topics with many kinds of ontologies, other than about humans. We also plan to extend our algorithm to deal with multiple entities on a page, or to make use of HTML tag information to improve the extraction accuracy of the system.

References

H. H. Chen, S. C. Tsai, and J. H. Tsai. 2000. Mining tables from large scale HTML texts. *18th International Conference on Computational Linguistics (COLING)*, pages 166–172.

J.M. Conroy, J.D. Schlesinger, and D.P. O'Leary. 2001. Using hmm and logistic regression to generate extract summaries for duc. *In Proceedings of ACM SIGIR Workshop on Text Summarization.*

F. Jelinek. 1998. Statistical methods for speech recognition. *MIT Press.*

W. Kraaij, M. Spitters, and M.v.d. Heijden. 2001. Combining a mixture language model and naive bayes for multi-document summarisation. *In Proceedings of ACM SIGIR Workshop on Text Summarization.*

S. Kurohashi and M. Nagao. 1998. Japanese morphological analysis system JUMAN version 3.5. *Department of Informatics, Kyoto University, (in Japanese).*

C.D. Manning and H. Schütze. 1999. Foundations of statistical natural language processing. *MIT Press.*

M. Yoshida, K. Torisawa, and J. Tsujii. 2001. Extracting ontologies from World Wide Web via HTML tables. *In Proceeding of Pacific Association for Computational Linguistics (PACLING)*, pages 332–341.